



ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно && видно?



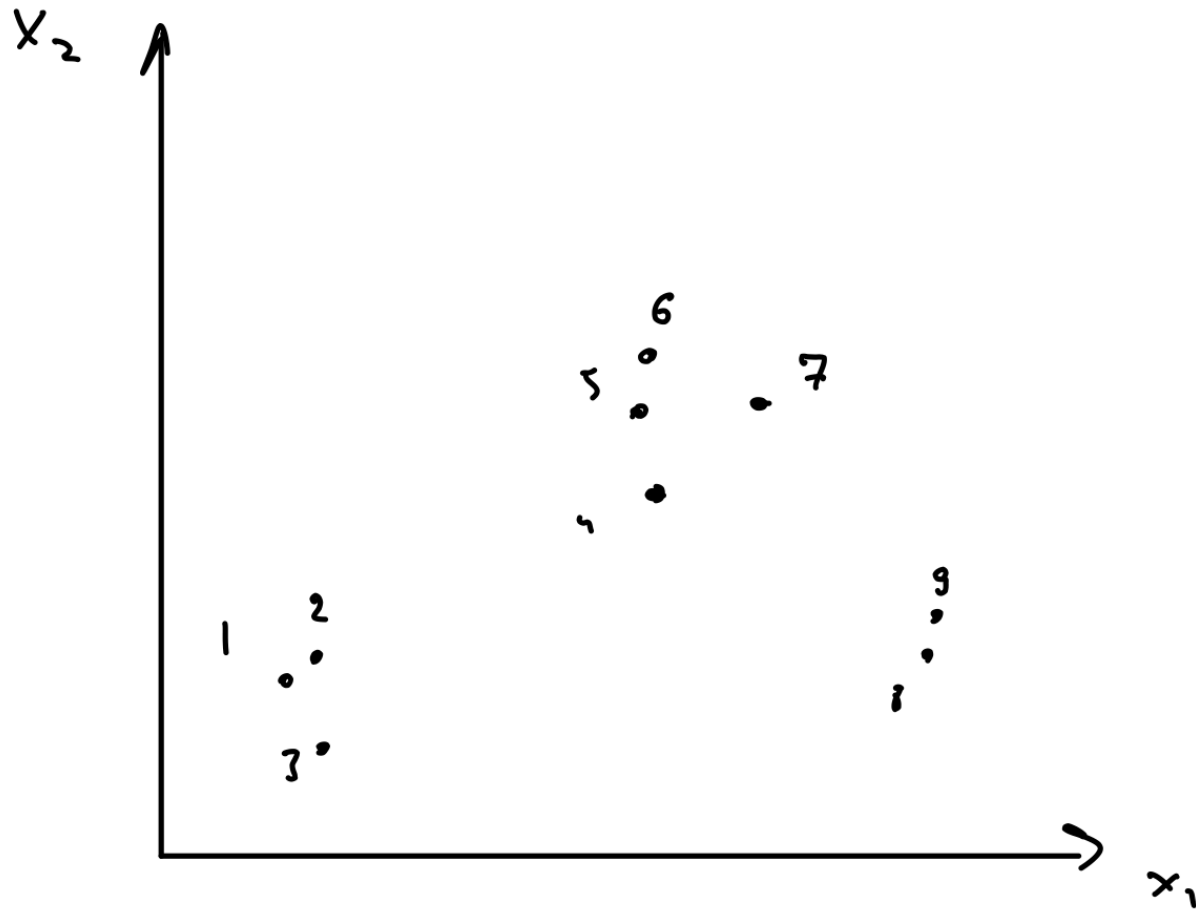
Напишите в чат, если есть проблемы!

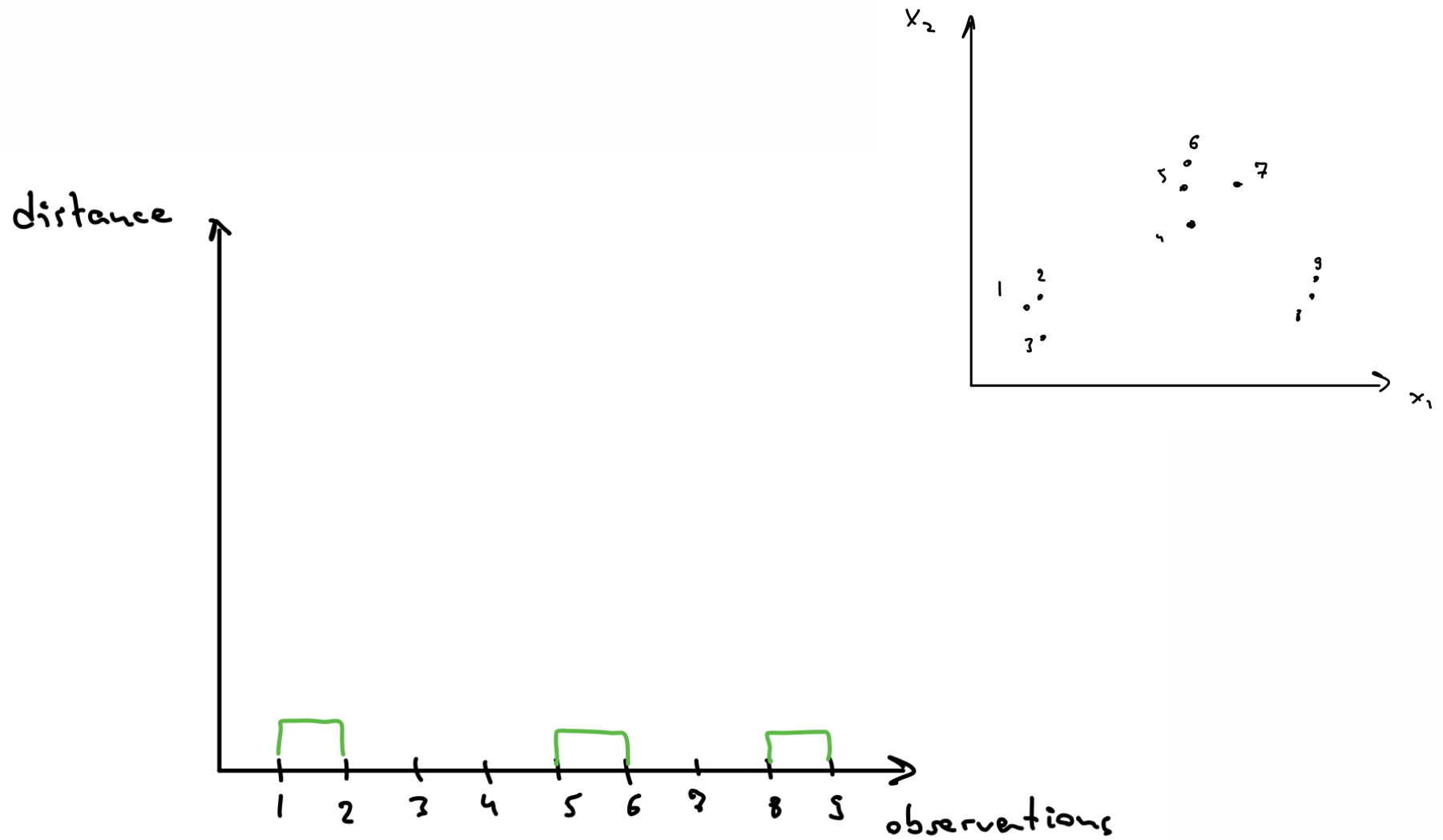
Ставьте ☐ + если все хорошо

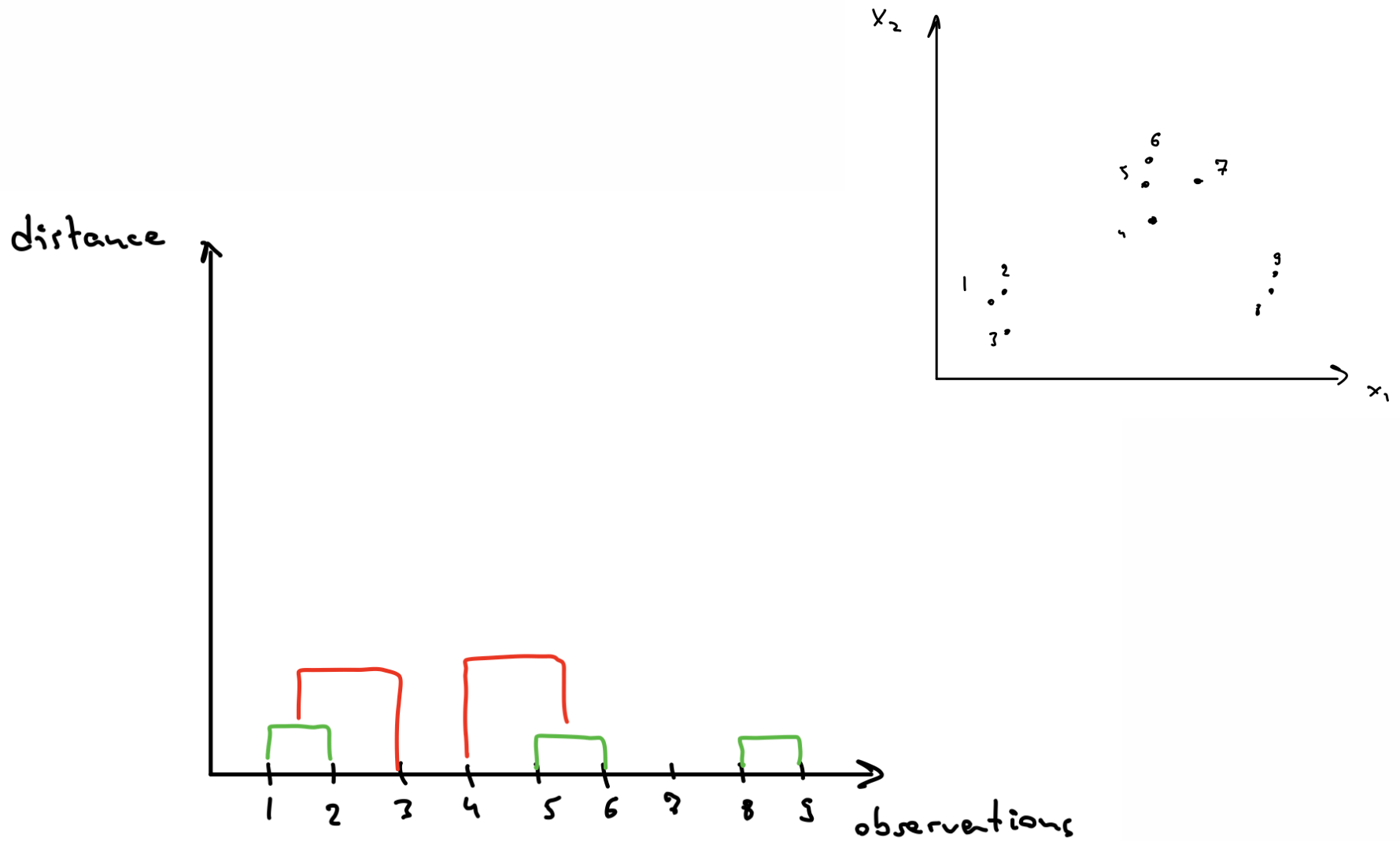
- Познакомиться с еще несколькими алгоритмами кластеризации
- Понять принципы работы алгоритмов и зачем они вообще нужны
- Узнать о методах настройки моделей

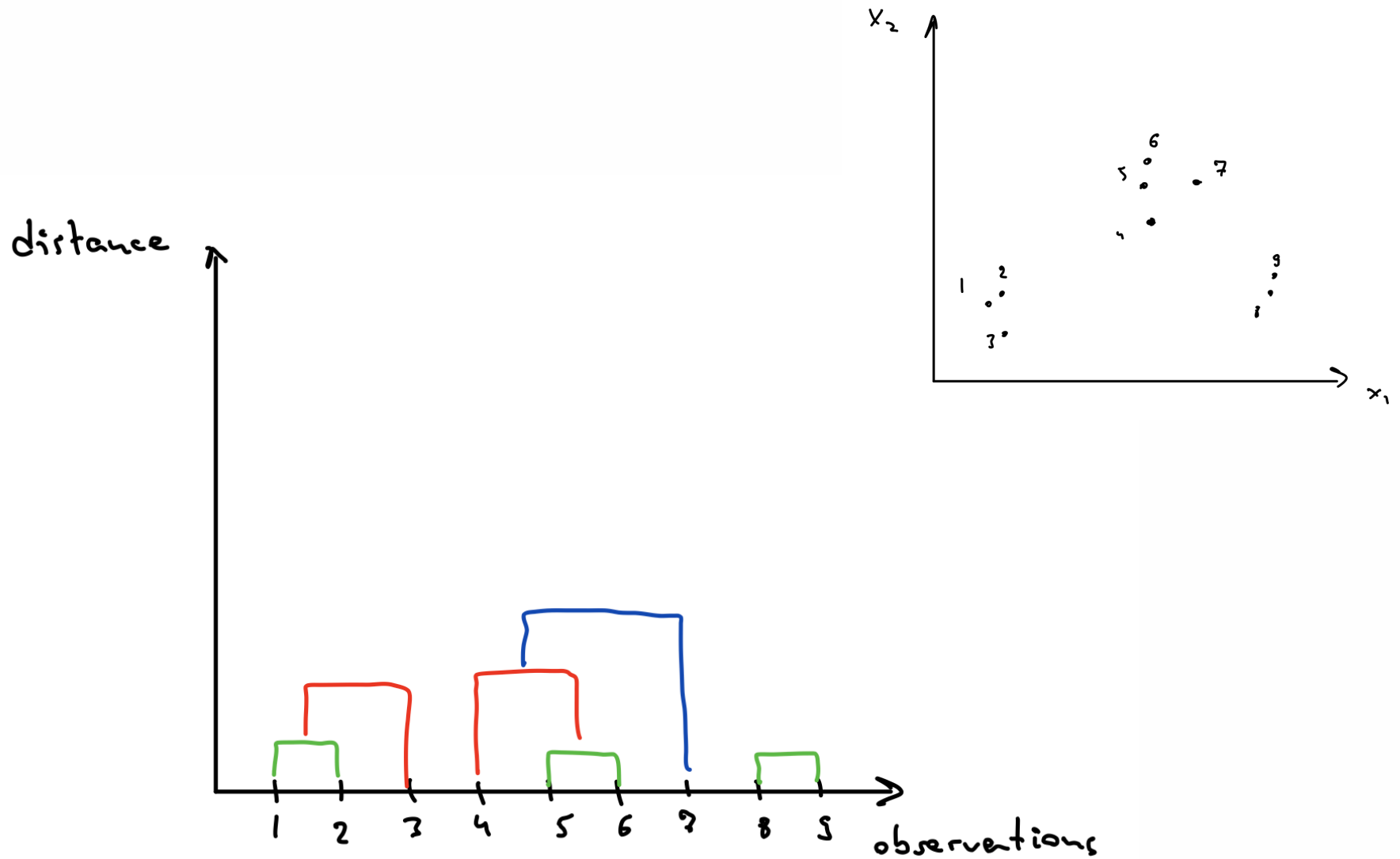
01

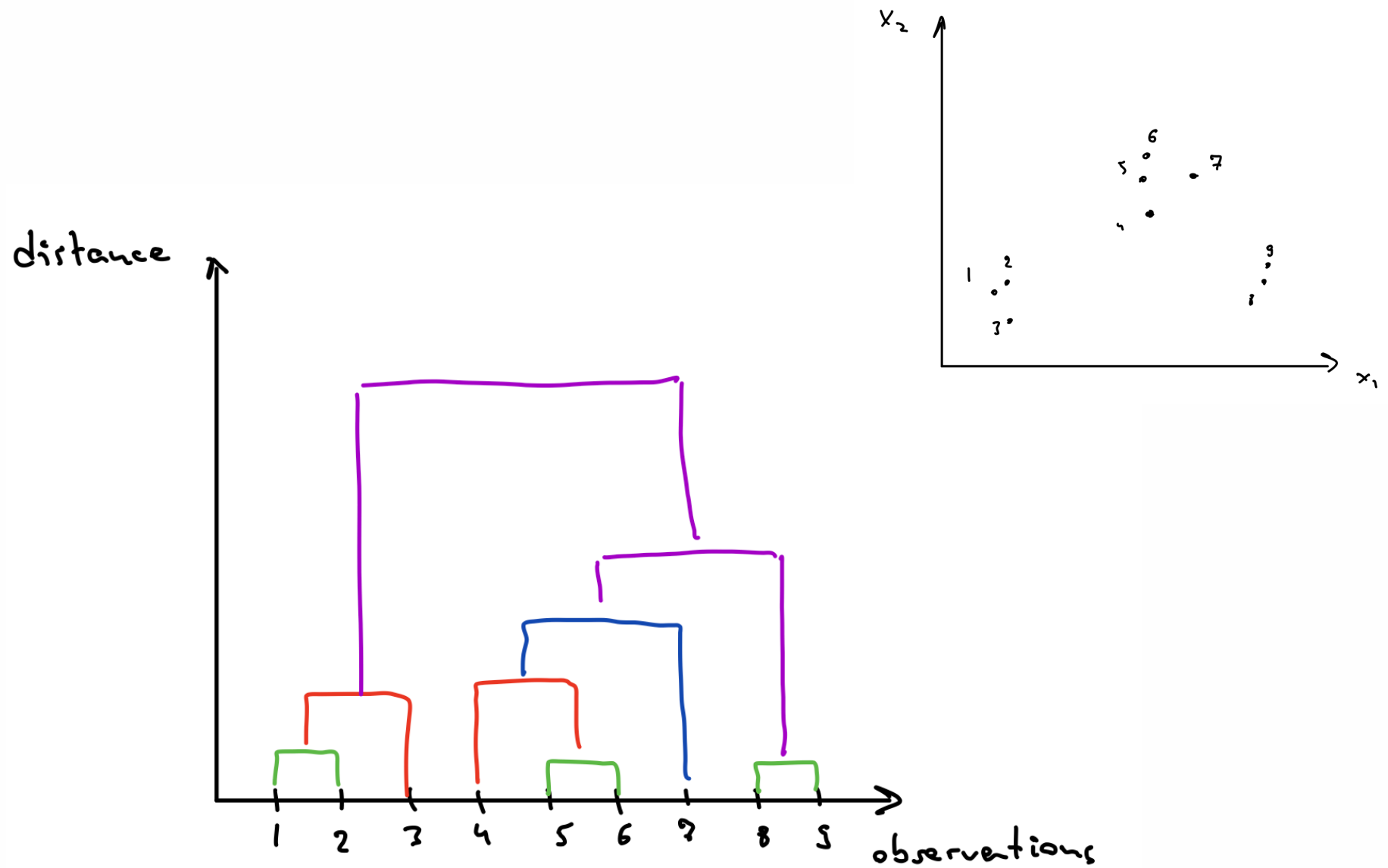
Иерархическая кластеризация

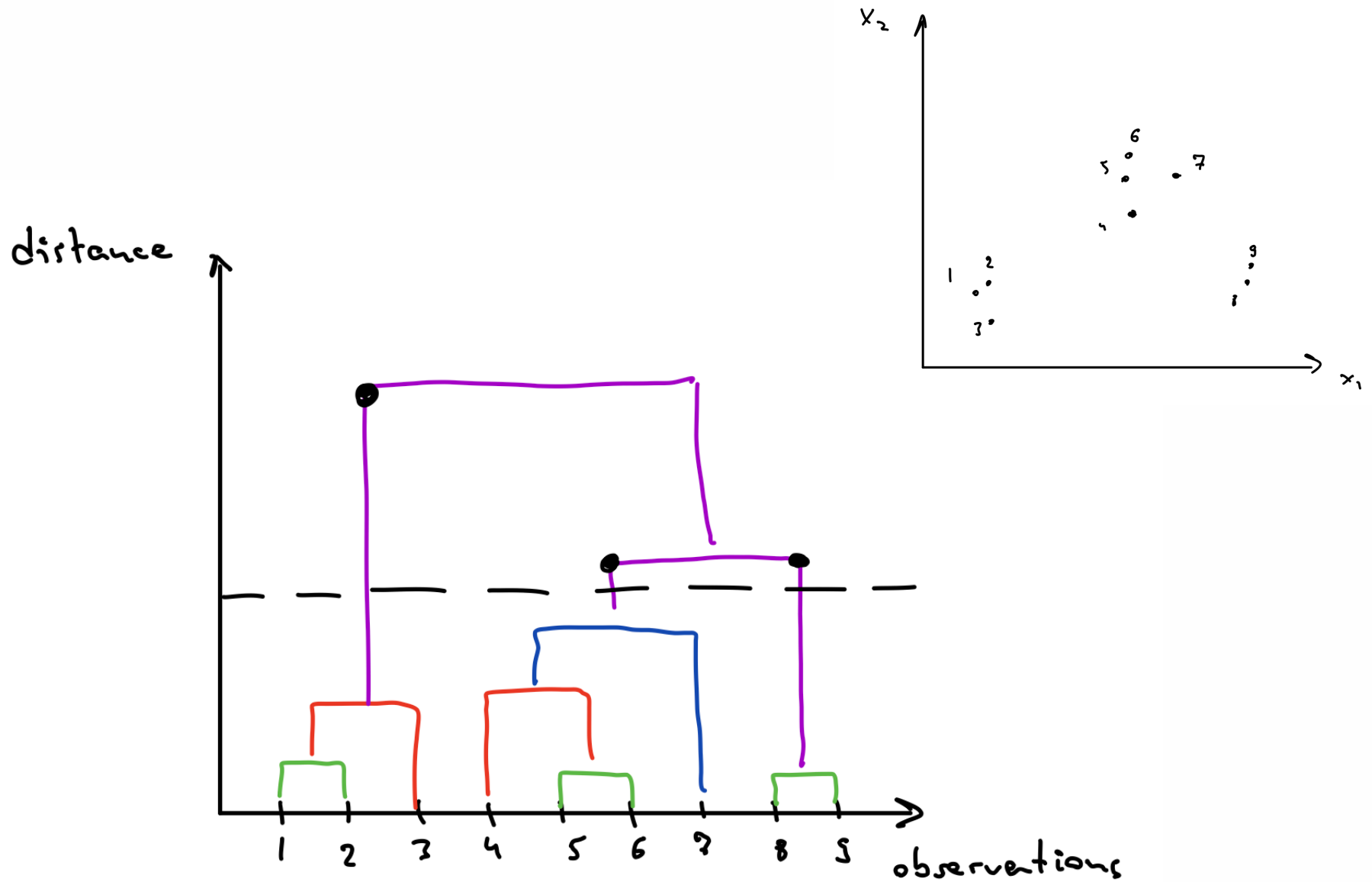












- Алгоритм:
- 1. Все наблюдения – отдельные кластеры

- Алгоритм:
- 1. Все наблюдения – отдельные кластеры
- 2. Считаем попарные расстояния и объединяем самые близкие точки в новые кластеры

- Алгоритм:
- 1. Все наблюдения – отдельные кластеры
- 2. Считаем попарные расстояния и объединяем самые близкие точки в новые кластеры
- 3. Снова считаем попарные расстояния между оставшимися точками и центрами новых кластеров

- Алгоритм:
- 1. Все наблюдения – отдельные кластеры
- 2. Считаем попарные расстояния и объединяем самые близкие точки в новые кластеры
- 3. Снова считаем попарные расстояния между оставшимися точками и центрами новых кластеров
- Объединяем/группируем, пока все точки не окажутся в одном кластере

- Преимущества:
- Более тонкая работа с кластеризацией – понимаем, какие объекты действительно ближе друг ко другу

- Преимущества:
- Более тонкая работа с кластеризацией – понимаем, какие объекты действительно ближе друг ко другу
- Можем визуально определять число кластеров

- Недостатки:
- Работает медленнее

- Недостатки:
- Работает медленнее
- По-прежнему Distance-based алгоритм, т.е., чувствителен к выбору метрики расстояния

02

DBSCAN

Density-based spatial clustering of applications with noise

<https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>

- Алгоритм:
- 1. Выбираем случайную точку и находим её соседей в заданной окрестности

- Алгоритм:
- 1. Выбираем случайную точку и находим её соседей в заданной окрестности
- 2. Если соседей меньше критического значения – называем выбросами

- Алгоритм:
- 1. Выбираем случайную точку и находим её соседей в заданной окрестности
- 2. Если соседей меньше критического значения – называем выбросами
- 3. Если нет – объединяем в «плотный» кластер и повторяем поиск соседей

- Алгоритм:
- 1. Выбираем случайную точку и находим её соседей в заданной окрестности
- 2. Если соседей меньше критического значения – называем выбросами
- 3. Если нет – объединяем в «плотный» кластер и повторяем поиск соседей
- 4. Если все плотные точки пройдены и помечены как посещенные – выбираем новую не посещенную точку и начинаем сначала

- Алгоритм:
- 1. Выбираем случайную точку и находим её соседей в заданной окрестности
- 2. Если соседей меньше критического значения – называем выбросами
- 3. Если нет – объединяем в «плотный» кластер и повторяем поиск соседей
- 4. Если все плотные точки пройдены и помечены как посещенные – выбираем новую не посещенную точку и начинаем сначала
- Повторяем, пока все точки не будут посещены

- Преимущества:
- Density-based (плотностной/вероятностный) метод – умеет в сложные формы кластеров
- Поиск выбросов и аномалий в данных (!)

- Недостатки:
- Сложный в настройке – очень чувствителен к параметру “плотности” ϵ



03

Итоги

- Иерархическая кластеризация – знаем зачем и почему
- DBSCAN - аналогично

Спасибо
за внимание!

